

Support Vector Machines

Nipun Batra

July 21, 2025

IIT Gandhinagar

Outline

Introduction and Motivation

Mathematical Foundation

SVM Formulation

Worked Example

Kernel Methods

- Kernel Motivation

- Kernel Examples

- Kernel Properties

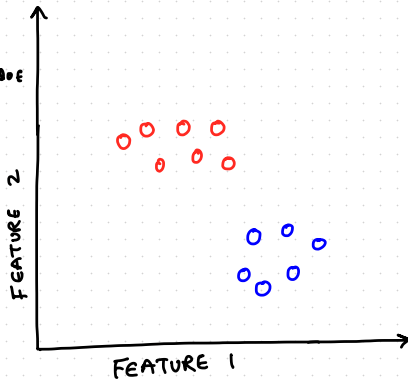
Summary

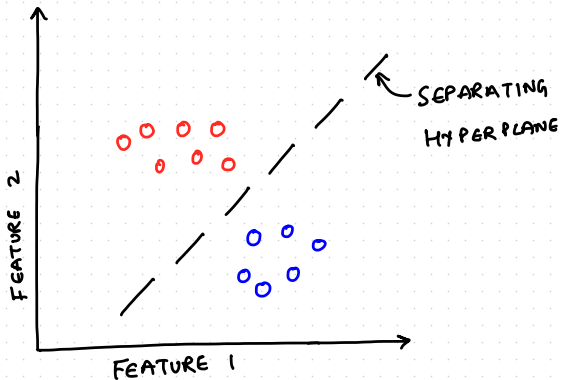
Introduction and Motivation

SUPPORT VECTOR MACHINES

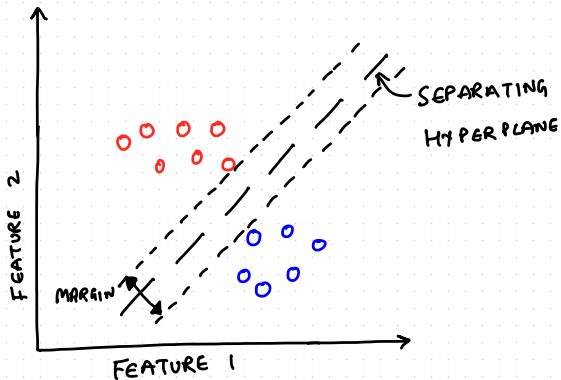
POPULAR BINARY

CLASSIFICATION TECHNIQUE

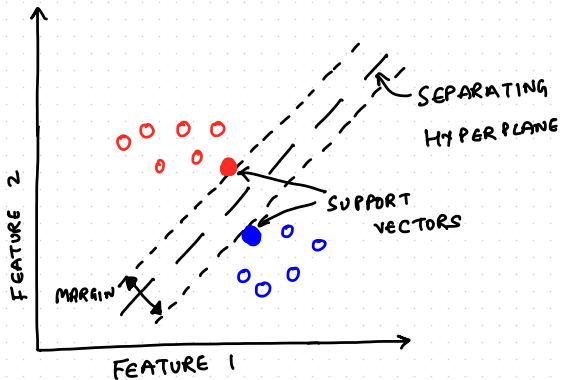




IDEA: DRAW A SEPARATING HYPER PLANE



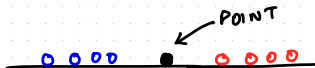
IDEA: MAXIMIZE THE MARGIN



SUPPORT VECTORS: POINTS ON BOUNDARY | MARGIN

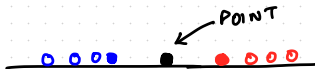
HYPERPLANE vs # DIMENSIONS

1D

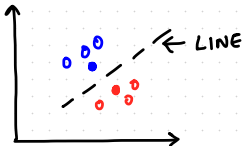


HYPERPLANE VS # DIMENSIONS

1D

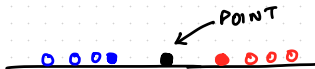


2D

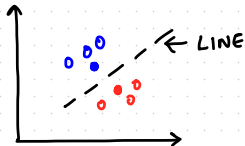


HYPERPLANE VS # DIMENSIONS

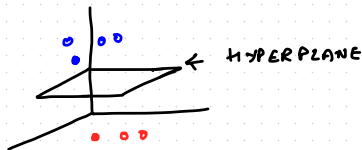
1D



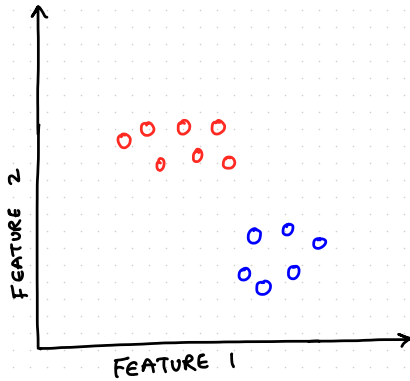
2D



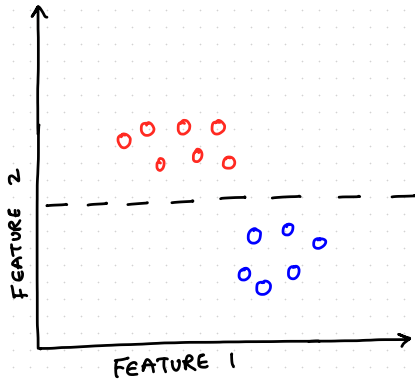
3D
(AND
MORE)



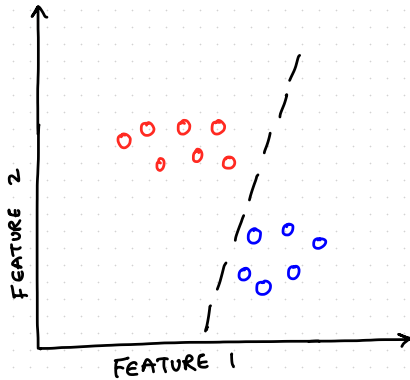
WHICH HYPER PLANE?



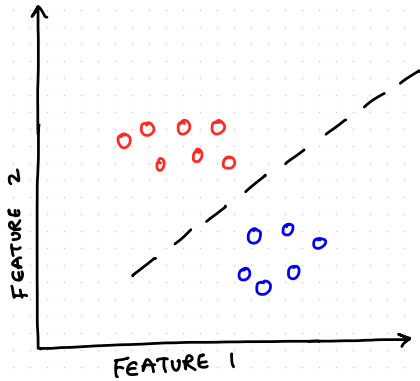
WHICH HYPER PLANE?



WHICH HYPER PLANE?

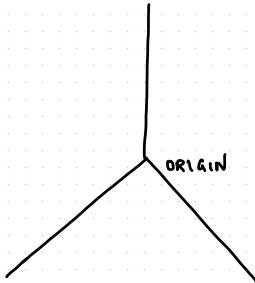


WHICH HYPER PLANE?

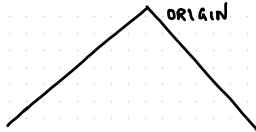


EQUATION OF HYPERPLANE

How to define?

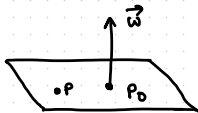


EQUATION OF HYPERPLANE

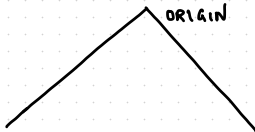


P : Any point on plane
 P_0 : One point on plane

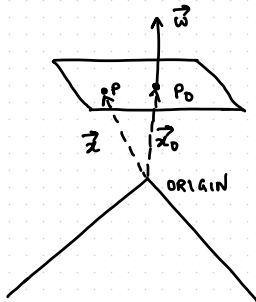
EQUATION OF HYPERPLANE



\vec{w} : \perp vector to
plane at P_0

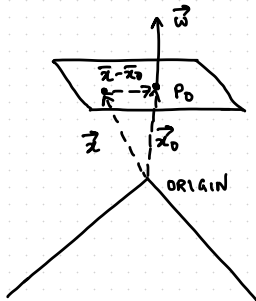


EQUATION OF HYPERPLANE



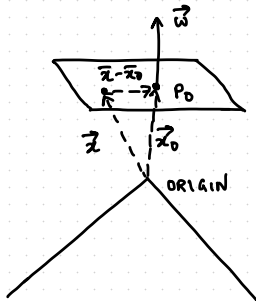
P and P_0 lie on plane

EQUATION OF HYPERPLANE



$\vec{w}^T (\vec{x} - \vec{x}_0) = 0$ lies on plane

EQUATION OF HYPERPLANE



$\vec{P}P_0 = \vec{x} - \vec{x}_0$ lies on plane

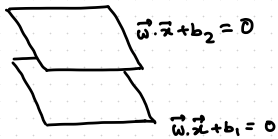
$$\Rightarrow \vec{w} \perp (\vec{x} - \vec{x}_0)$$

$$\text{or, } \vec{w} \cdot (\vec{x} - \vec{x}_0) = 0$$

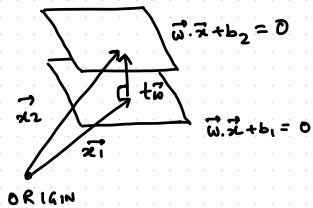
$$\text{or, } \vec{w} \cdot \vec{x} - \vec{w} \cdot \vec{x}_0 = 0$$

$$\text{or, } \boxed{\vec{w} \cdot \vec{x} + b = 0}$$

DISTANCE B/W || HYPER PLANES



DISTANCE B/W || HYPER PLANES



Mathematical Foundation

Distance between 2 parallel hyperplanes

Equation of two planes is:

$$\mathbf{w} \cdot \mathbf{x} + b_1 = 0$$

$$\mathbf{w} \cdot \mathbf{x} + b_2 = 0$$

Distance between 2 parallel hyperplanes

Equation of two planes is:

$$\mathbf{w} \cdot \mathbf{x} + b_1 = 0$$

$$\mathbf{w} \cdot \mathbf{x} + b_2 = 0$$

For a point \mathbf{x}_1 on plane 1 and \mathbf{x}_2 on plane 2, we have:

Distance between 2 parallel hyperplanes

Equation of two planes is:

$$\mathbf{w} \cdot \mathbf{x} + b_1 = 0$$

$$\mathbf{w} \cdot \mathbf{x} + b_2 = 0$$

For a point \mathbf{x}_1 on plane 1 and \mathbf{x}_2 on plane 2, we have:

$$\mathbf{x}_2 = \mathbf{x}_1 + t\mathbf{w}$$

$$D = |t\mathbf{w}| = |t||\mathbf{w}|$$

Distance between 2 parallel hyperplanes

Equation of two planes is:

$$\mathbf{w} \cdot \mathbf{x} + b_1 = 0$$

$$\mathbf{w} \cdot \mathbf{x} + b_2 = 0$$

For a point \mathbf{x}_1 on plane 1 and \mathbf{x}_2 on plane 2, we have:

$$\mathbf{x}_2 = \mathbf{x}_1 + t\mathbf{w}$$

$$D = |t\mathbf{w}| = |t||\mathbf{w}|$$

We can rewrite as follows:

Distance between 2 parallel hyperplanes

Equation of two planes is:

$$\mathbf{w} \cdot \mathbf{x} + b_1 = 0$$

$$\mathbf{w} \cdot \mathbf{x} + b_2 = 0$$

For a point \mathbf{x}_1 on plane 1 and \mathbf{x}_2 on plane 2, we have:

$$\mathbf{x}_2 = \mathbf{x}_1 + t\mathbf{w}$$

$$D = |t\mathbf{w}| = |t||\mathbf{w}|$$

We can rewrite as follows:

$$\mathbf{w} \cdot \mathbf{x}_2 + b_2 = 0$$

$$\Rightarrow \mathbf{w} \cdot (\mathbf{x}_1 + t\mathbf{w}) + b_2 = 0$$

Distance between 2 parallel hyperplanes

Equation of two planes is:

$$\mathbf{w} \cdot \mathbf{x} + b_1 = 0$$

$$\mathbf{w} \cdot \mathbf{x} + b_2 = 0$$

For a point \mathbf{x}_1 on plane 1 and \mathbf{x}_2 on plane 2, we have:

$$\mathbf{x}_2 = \mathbf{x}_1 + t\mathbf{w}$$

$$D = |t\mathbf{w}| = |t| \|\mathbf{w}\|$$

We can rewrite as follows:

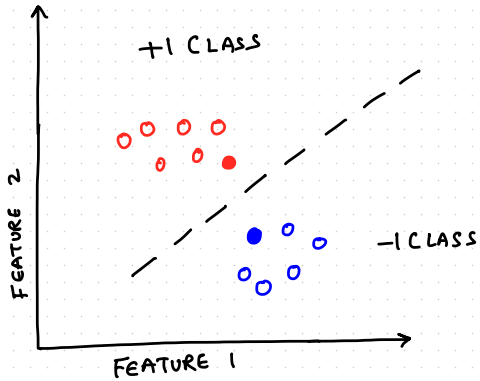
$$\mathbf{w} \cdot \mathbf{x}_2 + b_2 = 0$$

$$\Rightarrow \mathbf{w} \cdot (\mathbf{x}_1 + t\mathbf{w}) + b_2 = 0$$

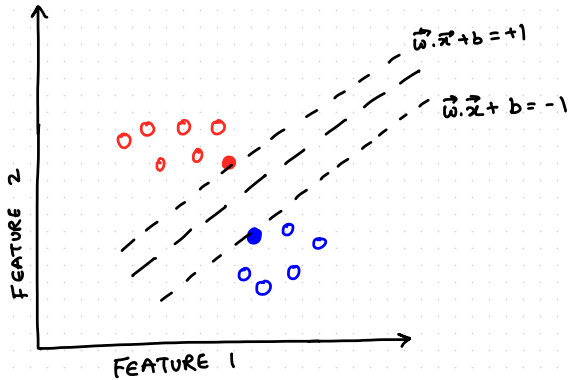
$$\Rightarrow \mathbf{w} \cdot \mathbf{x}_1 + t \|\mathbf{w}\|^2 + b_1 - b_1 + b_2 = 0 \Rightarrow t = \frac{b_1 - b_2}{\|\mathbf{w}\|^2} \Rightarrow D = t \|\mathbf{w}\| = \frac{b_1 - b_2}{\|\mathbf{w}\|}$$

SVM Formulation

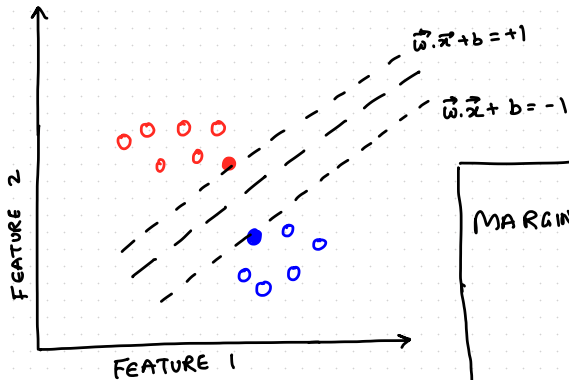
FORMULATION



FORMULATION

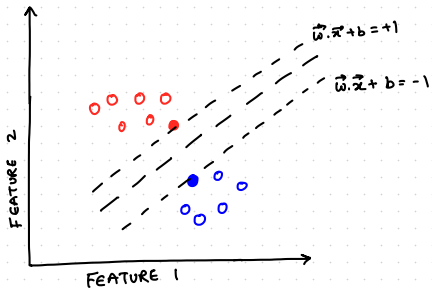


FORMULATION



$$\begin{aligned} \text{MARGIN} &= \frac{(b+1) - (b-1)}{\|\vec{w}\|} \\ &= \frac{2}{\|\vec{w}\|} \end{aligned}$$

FORMULATION



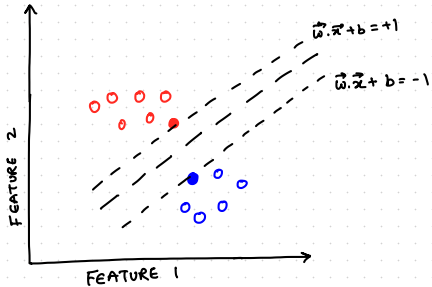
GOAL: MAXIMIZE MARGIN

$$\Rightarrow \text{MAXIMIZE } \frac{2}{\|\vec{w}\|}$$

$$\Rightarrow \text{MINIMIZE } \|\vec{w}\|$$

S.T. Correctly label points

FORMULATION



GOAL: MAXIMIZE MARGIN

$$\Rightarrow \text{MAXIMIZE } \frac{2}{\|\vec{w}\|}$$

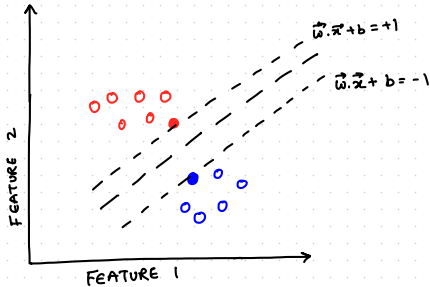
$$\Rightarrow \text{MINIMIZE } \|\vec{w}\|$$

S.T. Correctly label points

i.e. if $y_i = -1$
 $\vec{w} \cdot \vec{x} + b \leq -1$

if $y_i = +1$
 $\vec{w} \cdot \vec{x} + b \geq +1$

FORMULATION



GOAL: MAXIMIZE MARGIN

$$\Rightarrow \text{MAXIMIZE } \frac{2}{\|\vec{w}\|}$$

$$\Rightarrow \boxed{\text{MINIMIZE } \|\vec{w}\|}$$

S.T. correctly label points

$$\text{i.e. if } y_i = -1 \\ \vec{w} \cdot \vec{x} + b \leq -1$$

$$\text{if } y_i = +1 \\ \vec{w} \cdot \vec{x} + b \geq +1$$

$$\boxed{y_i (\vec{w} \cdot \vec{x} + b) \geq 1}$$

Primal Formulation

Objective

$$\begin{aligned} & \text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{subject to } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \forall i \end{aligned}$$

Primal Formulation

Objective

$$\begin{aligned} & \text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{subject to } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \forall i \end{aligned}$$

Q) What is $\|\mathbf{w}\|$?

Primal Formulation

Objective

$$\begin{aligned} & \text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{subject to } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \forall i \end{aligned}$$

Q) What is $\|\mathbf{w}\|$?

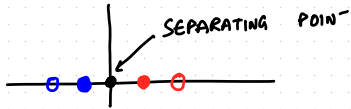
$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

$$\|\mathbf{w}\| = \sqrt{\mathbf{w}^\top \mathbf{w}}$$

$$= \sqrt{\begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}}$$

Worked Example

EXAMPLE (IN 1D)



Simple Exercise

$$\begin{bmatrix} x & y \\ 1 & 1 \\ 2 & 1 \\ -1 & -1 \\ -2 & -1 \end{bmatrix}$$

Separating Hyperplane: $\mathbf{w} \cdot \mathbf{x} + b = 0$

Simple Exercise

$$y_i(w_i x_i + b) \geq 1$$

$$\begin{bmatrix} x_1 & y \\ 1 & 1 \\ 2 & 1 \\ -1 & -1 \\ -2 & -1 \end{bmatrix}$$

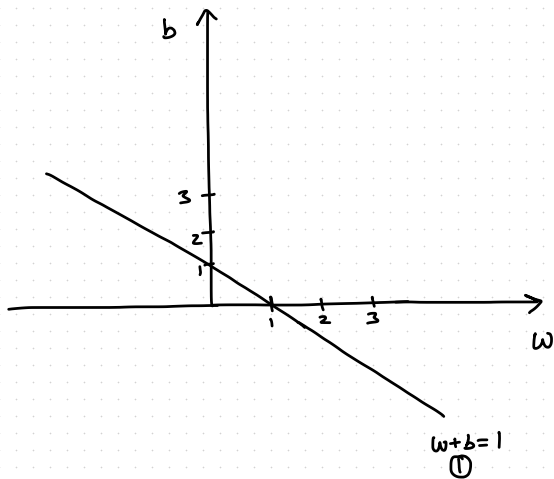
$$\Rightarrow y_i(w_i x_i + b) \geq 1$$

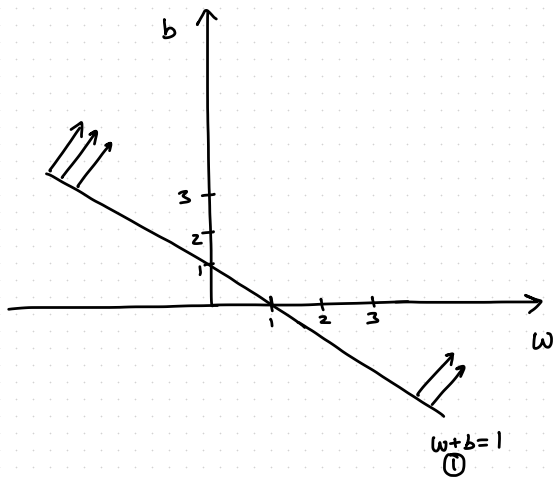
$$\Rightarrow 1(w_1 + b) \geq 1$$

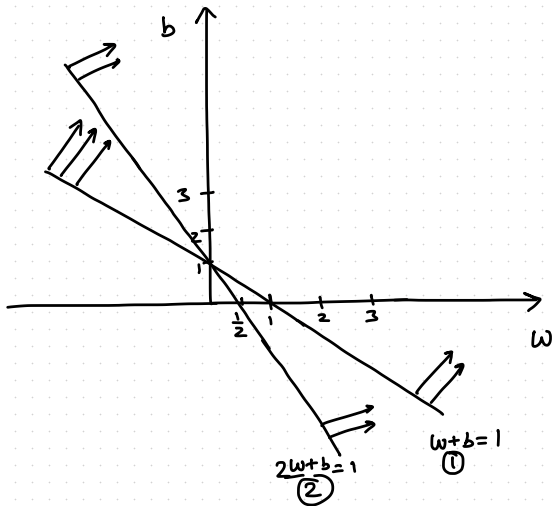
$$\Rightarrow 1(2w_1 + b) \geq 1$$

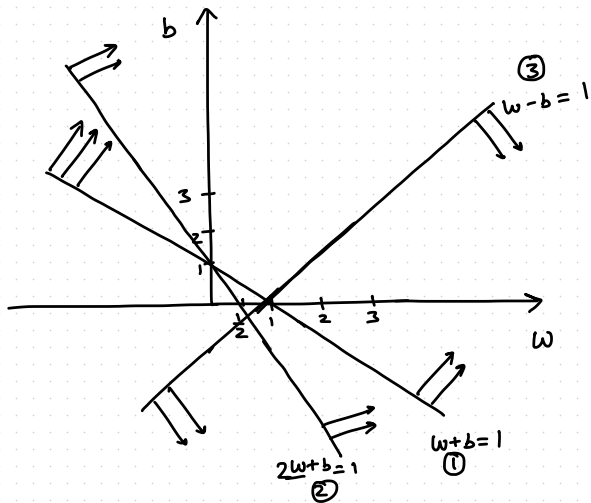
$$\Rightarrow -1(-w_1 + b) \geq 1$$

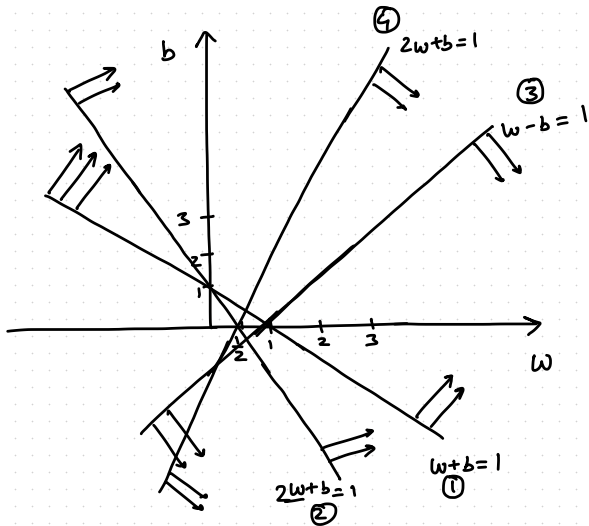
$$\Rightarrow -1(-2w_1 + b) \geq 1$$

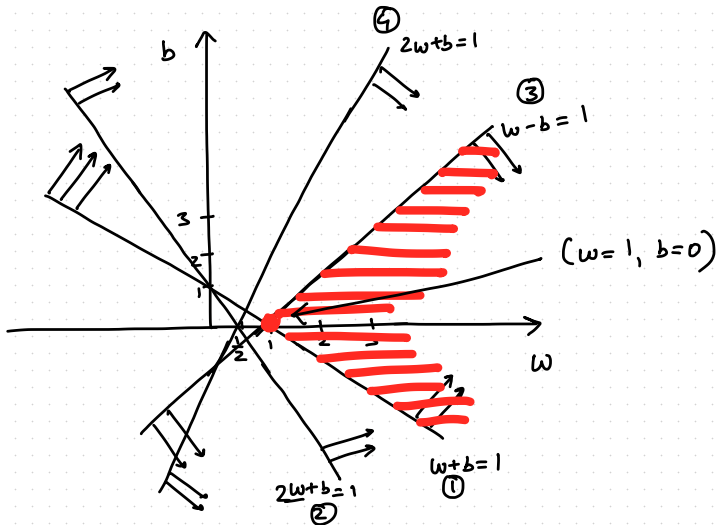












Simple Exercise

$$w_{min} = 1, b = 0$$

$$w.x + b = 0$$

$$x = 0$$

Simple Exercise

Minimum values satisfying constraints $\Rightarrow w = 1$ and $b = 0$

\therefore Max margin classifier $\Rightarrow x = 0$

Primal Formulation is a Quadratic Program

Generally;

\Rightarrow Minimize Quadratic(x)

\Rightarrow such that, Linear(x)

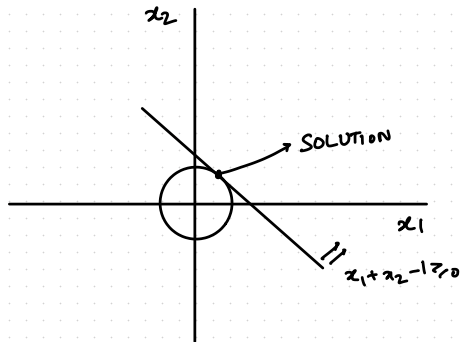
Question

$$x = (x_1, x_2)$$

$$\text{minimize } \frac{1}{2} \|x\|^2$$

$$: x_1 + x_2 - 1 \geq 0$$

MINIMIZE QUADRATIC
S.t. LINEAR



Converting to Dual Problem

Primal \Rightarrow Dual Conversion using Lagrangian multipliers

$$\begin{aligned} \text{Minimize } & \frac{1}{2} \|\bar{w}\|^2 \\ \text{s.t. } & y_i(\bar{w} \cdot x_i + b) \geq 1 \\ & \forall i \end{aligned}$$

$$L(\bar{w}, b, \alpha_1, \alpha_2, \dots, \alpha_n) = \frac{1}{2} \sum_{i=1}^d w_i^2 - \sum_{i=1}^N \alpha_i (y_i(\bar{w} \cdot \bar{x}_i + b) - 1) \quad \forall \alpha_i \geq 0$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^n \alpha_i y_i = 0$$

Converting to Dual Problem

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \bar{\mathbf{w}} - \sum_{i=1}^n \alpha_i y_i \bar{\mathbf{x}}_i = 0$$

$$\bar{\mathbf{w}} = \sum_{i=1}^N \alpha_i y_i \bar{\mathbf{x}}_i$$

$$L(\bar{\mathbf{w}}, b, \alpha_1, \alpha_2, \dots, \alpha_n) = \frac{1}{2} \sum_{i=1}^d w_i^2 - \sum_{i=1}^N \alpha_i (y_i (\bar{\mathbf{w}} \cdot \bar{\mathbf{x}}_i + b) - 1)$$

$$= \frac{1}{2} \|\bar{\mathbf{w}}\|^2 - \sum_{i=1}^N \alpha_i y_i \bar{\mathbf{w}} \cdot \bar{\mathbf{x}}_i - \sum_{i=1}^N \alpha_i y_i b + \sum_{i=1}^N \alpha_i$$

$$= \sum_{i=1}^N \alpha_i + \frac{(\sum_i \alpha_i y_i \bar{\mathbf{x}}_i) (\sum_j \alpha_j y_j \bar{\mathbf{x}}_j)}{2} - \sum_i \alpha_i y_i \left(\sum_j \alpha_j y_j \bar{\mathbf{x}}_j \right) \bar{\mathbf{x}}_i$$

Converting to Dual Problem

$$L(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \bar{x}_i \cdot \bar{x}_j$$

$$\begin{array}{ll} \text{Minimize } \|\bar{w}\|^2 & \Rightarrow \text{Maximize } L(\alpha) \\ \text{s.t} & \text{s.t} \\ y_i (\bar{w}, x_i + b) \geq 1 & \sum_{i=1}^N \alpha_i y_i = 0 \quad \forall \alpha_i \geq 0 \end{array}$$

Question

Question:

$\alpha_i (y_i (\bar{w} \cdot \bar{x}_i + b) - 1) = 0 \quad \forall i$ as per KKT slackness

What is α_i for support vector points?

Answer: For support vectors,

$$\bar{w} \cdot \bar{x}_i + b = -1 \quad (+ve \text{ class})$$

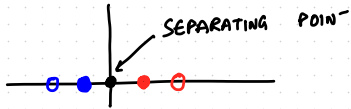
$$\bar{w} \cdot \bar{x}_i + b = +1 \quad (+ve \text{ class})$$

$$y_i (\bar{w} \cdot \bar{x}_i + b) - 1 = 0 \quad \text{for } i = \{\text{support vector points}\}$$

$$\therefore \alpha_i \text{ where } i \in \{\text{support vector points}\} \neq 0$$

For all non-support vector points $\alpha_i = 0$

EXAMPLE (IN 1D)



Revisiting the Simple Example

$$\begin{bmatrix} x_1 & y \\ 1 & 1 \\ 2 & 1 \\ -1 & -1 \\ -2 & -1 \end{bmatrix}$$

$$L(\alpha) = \sum_{i=1}^4 \alpha_i - \frac{1}{2} \sum_{i=1}^4 \sum_{j=1}^4 \alpha_i \alpha_j y_i y_j \bar{x}_i \bar{x}_j \quad \alpha_i \geq 0$$
$$\sum \alpha_i y_i = 0 \quad \alpha_i (y_i (\bar{w} \cdot \bar{x}_i + b - 1)) = 0$$

Revisiting the Simple Example

$$\begin{aligned} L(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = & \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \\ & - \frac{1}{2} \{ \alpha_1 \alpha_1 \times (1 * 1) \times (1 * 1) \\ & + \\ & \alpha_1 \alpha_2 \times (1 * 1) \times (1 * 2) \\ & + \\ & \alpha_1 \alpha_3 \times (1 * -1) \times (1 * 1) \\ & \dots \\ & \alpha_4 \alpha_4 \times (-1 * -1) \times (-2 * -2) \} \end{aligned}$$

How to Solve? \Rightarrow Use the QP Solver!!

Revisiting the Simple Example

For the trivial example,

We know that only $x = \pm 1$ will take part in the constraint actively.

Thus, $\alpha_2, \alpha_4 = 0$

By symmetry, $\alpha_1 = \alpha_3 = \alpha$ (say)

$$\& \sum y_i \alpha_i = 0$$

$$L(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = 2\alpha$$

$$\begin{aligned} & - \frac{1}{2} \{ \alpha^2(1)(-1)(1)(-1) \\ & \quad + \alpha^2(-1)(1)(-1)(1) \\ & \quad + \alpha^2(1)(1)(1)(1) + \alpha^2(-1)(-1)(-1)(-1) \\ & \quad \} \end{aligned}$$

$$\underset{\alpha}{\text{Maximize}} \quad 2\alpha - \frac{1}{2}(4\alpha^2)$$

Revisiting the Simple Example

$$\frac{\partial}{\partial \alpha} (2\alpha - 2\alpha^2) = 0 \Rightarrow 2 - 4\alpha = 0$$

$$\Rightarrow \alpha = 1/2$$

$$\therefore \alpha_1 = 1/2 \quad \alpha_2 = 0; \quad \alpha_3 = 1/2 \quad \alpha_4 = 0$$

$$\begin{aligned} \mathbf{w} &= \sum_{i=1}^N \alpha_i y_i \bar{x}_i = 1/2 \times 1 \times 1 + 0 \times 1 \times 2 \\ &\quad + 1/2 \times -1 \times -1 + 0 \times -1 \times -2 \\ &= 1/2 + 1/2 = 1 \end{aligned}$$

Revisiting the Simple Example

Finding b :

For the support vectors we have,

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 = 0$$

$$\text{or, } y_i (\bar{w} \cdot \bar{x}_1 + b) = 1$$

$$\text{or, } y_i^2 (\bar{w} \cdot \bar{x}_i + b) = y_i$$

$$\text{or, } \bar{w} \cdot \bar{x}_i + b = y_i \quad (\because y_i^2 = 1)$$

$$\text{or, } b = y_i - \bar{w} \cdot \bar{x}_i$$

$$\text{In practice, } b = \frac{1}{N_{SV}} \sum_{i=1}^{N_{SV}} (y_i - \bar{w} \bar{x}_i)$$

Obtaining the Solution

$$\begin{aligned} b &= \frac{1}{2}\{(1 - (1)(1)) + (-1 - (1)(-1))\} \\ &= \frac{1}{2}\{0 + 0\} = 0 \\ &= 0 \\ \therefore w &= 1 \text{ \& } b = 0 \end{aligned}$$

Making Predictions

$$\hat{y}(x_i) = \text{SIGN}(w \cdot x_i + b)$$

For $x_{\text{test}} = 3$; $\hat{y}(3) = \text{SIGN}(1 \times 3 + 0) = +\text{ve class}$

Making Predictions

Alternatively,

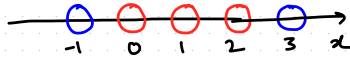
$$\begin{aligned}\hat{\mathbf{y}}(\mathbf{x}_{\text{test}}) &= \text{sign}(\mathbf{w} \cdot \mathbf{x}_{\text{test}} + b) \\ &= \text{sign}\left(\sum_{i=1}^{N_{\text{SV}}} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x}_{\text{test}} + b\right)\end{aligned}$$

In our example,

$$\alpha_1 = 1/2; \alpha_2 = 0; \quad \alpha_3 = 1/2; \alpha_4 = 0$$

$$\begin{aligned}\hat{\mathbf{y}}(3) &= \text{sign}\left(\frac{1}{2} \times 1 \times (1 \times 3) + 0 + \frac{1}{2} \times (-1) \times (-1 \times 3) + 0\right) \\ &= \text{sign}\left(\frac{6}{2}\right) = \text{sign}(3) = +1\end{aligned}$$

Kernel Methods



ORIGINAL DATA
IN R

Non-Linearly Separable Data

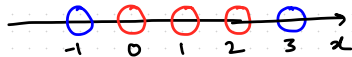
- Data is not linearly separable in \mathbb{R}^d .

Non-Linearly Separable Data

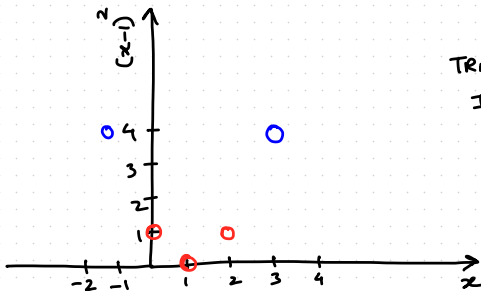
- Data is not linearly separable in \mathbb{R}^d .
- Can we still use SVM?

Non-Linearly Separable Data

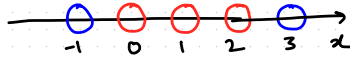
- Data is not linearly separable in \mathbb{R}^d .
- Can we still use SVM?
- Yes! Project data to a higher dimensional space.



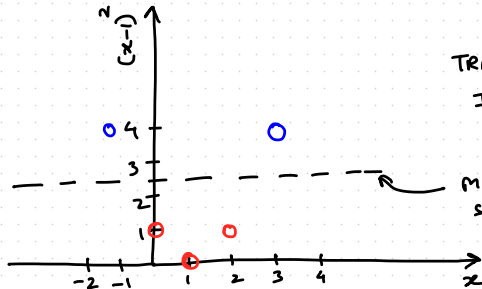
ORIGINAL DATA
IN \mathbb{R}



TRANSFORMED DATA
IN \mathbb{R}^2

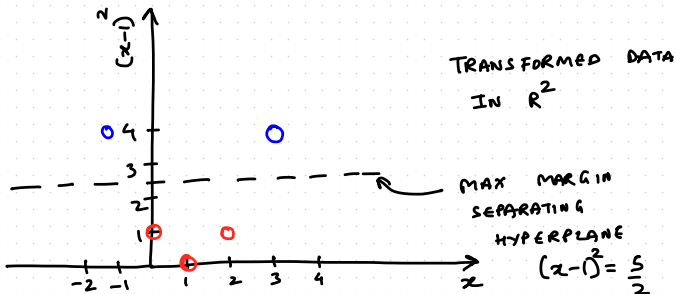
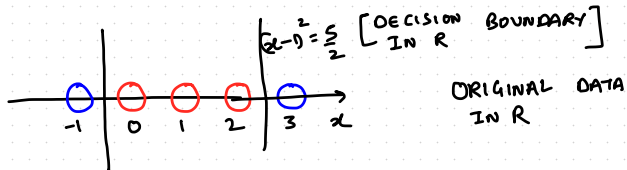


ORIGINAL DATA
IN \mathbb{R}

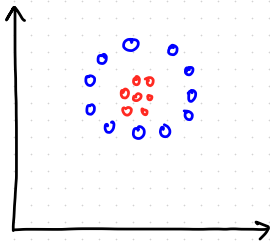


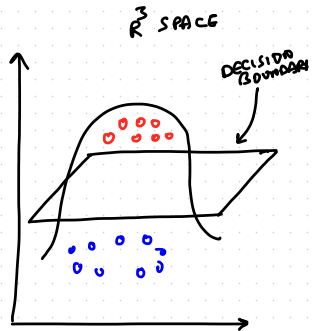
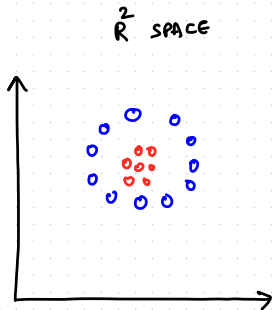
TRANSFORMED DATA
IN \mathbb{R}^2

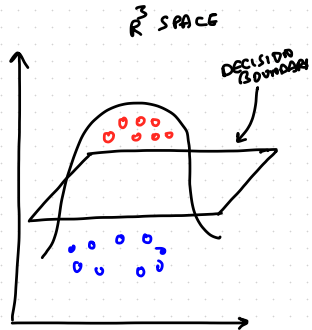
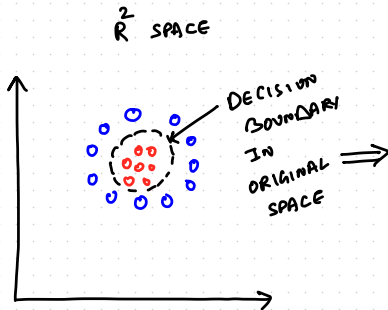
MAX MARGIN
SEPARATING
HYPERPLANE
 $(x-1)^2 = \frac{5}{2}$



\mathbb{R}^2 SPACE







Projection/Transformation Function

$$\phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$$

where, d = original dimension

D = new dimension

In our example:

$$d = 1; D = 2$$

From Linear to Kernel SVM

Linear SVM:

Maximize

$$L(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

such that constraints are satisfied.



Transformation (ϕ)



$$L(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

Steps

1. Compute $\phi(\mathbf{x})$ for each point

$$\phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$$

Q. If $D \gg d$

Both steps are expensive!

Steps

1. Compute $\phi(\mathbf{x})$ for each point

$$\phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$$

2. Compute dot products over \mathbb{R}^D space

Q. If $D \gg d$

Both steps are expensive!

The Kernel Trick

Brilliant idea: Can we compute $K(\mathbf{x}_i, \mathbf{x}_j)$ such that:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

Without explicitly computing ϕ !

- $K(\mathbf{x}_i, \mathbf{x}_j)$: Simple function in original space

Result: Get non-linear classification power without computational cost!

The Kernel Trick

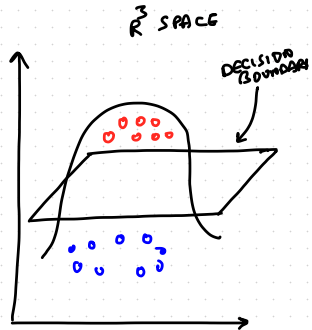
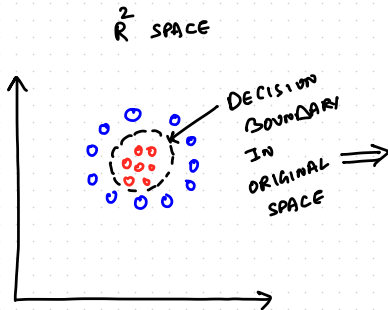
Brilliant idea: Can we compute $K(\mathbf{x}_i, \mathbf{x}_j)$ such that:

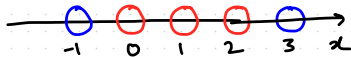
$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

Without explicitly computing ϕ !

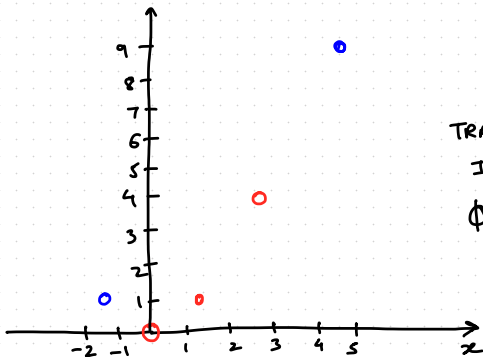
- $K(\mathbf{x}_i, \mathbf{x}_j)$: Simple function in original space
- $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$: Complex dot product in high-dimensional space

Result: Get non-linear classification power without computational cost!



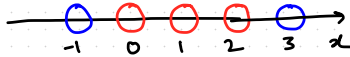


ORIGINAL DATA
IN \mathbb{R}

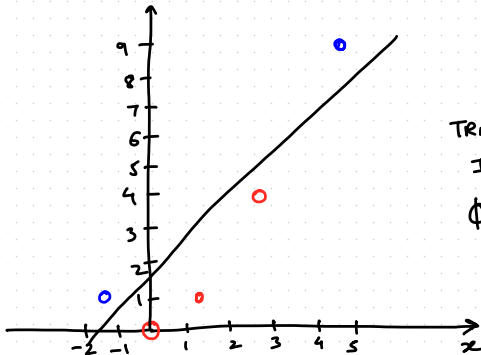


TRANSFORMED DATA
IN \mathbb{R}^2

$$\phi(x) = \begin{bmatrix} \sqrt{2} x \\ x^2 \end{bmatrix}$$

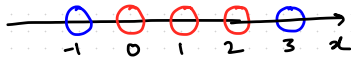


ORIGINAL DATA
IN \mathbb{R}

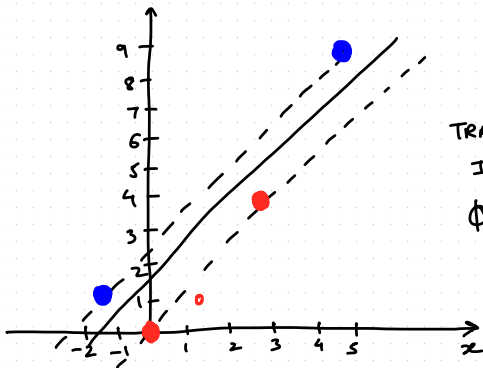


TRANSFORMED DATA
IN \mathbb{R}^2

$$\phi(x) = \begin{bmatrix} \sqrt{2}x \\ x^2 \end{bmatrix}$$

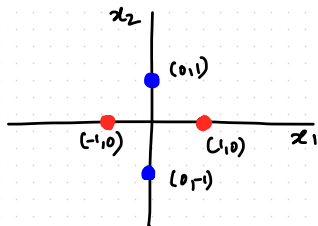


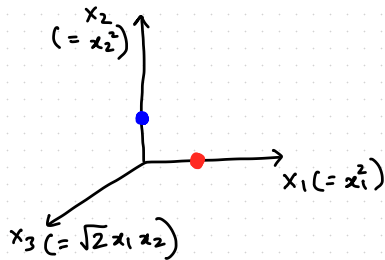
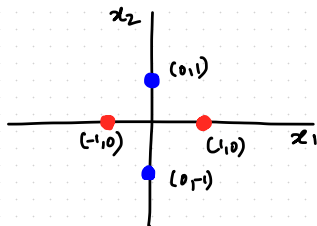
ORIGINAL DATA
IN \mathbb{R}

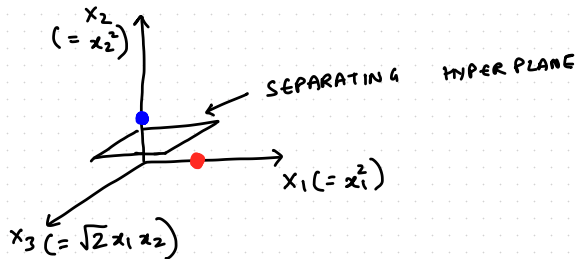
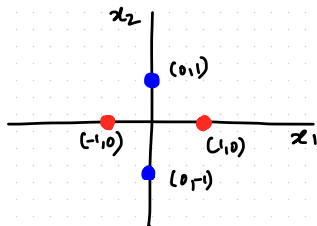


TRANSFORMED DATA
IN \mathbb{R}^2

$$\phi(x) = \begin{bmatrix} \sqrt{2}x \\ x^2 \end{bmatrix}$$







Q) Why did we use dual form?

Kernels again!!

Primal form doesn't allow for the kernel trick

$K(\mathbf{x}_1, \mathbf{x}_2)$ in dual and compute $\phi(\mathbf{x})$ and then dot product in D dimensions

Gram Matrix: (Positive Semi-Definite)

$$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i \cdot \mathbf{x}_j)^2$$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	24	8	0	0	8	24	48
x_2	8	1	0	-1	0	...	
x_3	0
x_4	0						
x_5	8						
x_6	24						
x_7	48						

Most frequently used kernels:

1. **Linear:** $K(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1 \cdot \mathbf{x}_2$

Parameters:

Most frequently used kernels:

1. **Linear:** $K(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1 \cdot \mathbf{x}_2$
2. **Polynomial:** $K(\mathbf{x}_1, \mathbf{x}_2) = (c + \mathbf{x}_1 \cdot \mathbf{x}_2)^d$

Parameters:

Most frequently used kernels:

1. **Linear:** $K(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1 \cdot \mathbf{x}_2$
2. **Polynomial:** $K(\mathbf{x}_1, \mathbf{x}_2) = (c + \mathbf{x}_1 \cdot \mathbf{x}_2)^d$
3. **RBF (Gaussian):** $K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma \|\mathbf{x}_1 - \mathbf{x}_2\|^2)$

Parameters:

Most frequently used kernels:

1. **Linear:** $K(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1 \cdot \mathbf{x}_2$
2. **Polynomial:** $K(\mathbf{x}_1, \mathbf{x}_2) = (c + \mathbf{x}_1 \cdot \mathbf{x}_2)^d$
3. **RBF (Gaussian):** $K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma \|\mathbf{x}_1 - \mathbf{x}_2\|^2)$

Parameters:

- c : constant term, d : degree (polynomial)

Most frequently used kernels:

1. **Linear:** $K(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1 \cdot \mathbf{x}_2$
2. **Polynomial:** $K(\mathbf{x}_1, \mathbf{x}_2) = (c + \mathbf{x}_1 \cdot \mathbf{x}_2)^d$
3. **RBF (Gaussian):** $K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma \|\mathbf{x}_1 - \mathbf{x}_2\|^2)$

Parameters:

- c : constant term, d : degree (polynomial)
- γ : bandwidth parameter (RBF)

Kernel Example: Polynomial Kernel

Question: For $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, what is the feature space for $K(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{x} \cdot \mathbf{z})^3$?

Given: $\mathbf{x} \in \mathbb{R}^2$, find dimension of $\phi(\mathbf{x})$

Expansion:

$$\begin{aligned} K(\mathbf{x}, \mathbf{z}) &= (1 + x_1 z_1 + x_2 z_2)^3 \\ &= \text{all terms of degree } \leq 3 \\ &= \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \end{aligned}$$

Feature map: $\phi(\mathbf{x}) =$

$$[1, \sqrt{3}x_1, \sqrt{3}x_2, \sqrt{3}x_1^2, \sqrt{3}x_2^2, \sqrt{6}x_1x_2, x_1^3, x_2^3, \sqrt{3}x_1^2x_2, \sqrt{3}x_1x_2^2]$$

Answer: $\phi(\mathbf{x}) \in \mathbb{R}^{10}$

RBF Kernel: Infinite Dimensions

Question: What is the dimensionality of RBF kernel feature space?

RBF Kernel:

$$\begin{aligned}K(x, z) &= \exp(-\gamma \|x - z\|^2) \\&= \exp(-\gamma (x - z)^2)\end{aligned}$$

Key insight: Using Taylor series expansion

$$\exp(\alpha) = \sum_{n=0}^{\infty} \frac{\alpha^n}{n!} = 1 + \alpha + \frac{\alpha^2}{2!} + \frac{\alpha^3}{3!} + \dots$$

Result: RBF kernel corresponds to ∞ -dimensional feature space!

Amazing: Infinite-dimensional classification with finite computation!

Does RBF Involve Dot Product in Lower-Dimensional Space?

Question: Can we see the original dot product in RBF kernel?

Assuming \mathbf{x} is a one-dimensional vector, we can rewrite the RBF kernel as:

$$K(x, z) = \exp(-\gamma \|x - z\|^2) = \exp(-\gamma (x - z)^2)$$

Does RBF Involve Dot Product in Lower-Dimensional Space?

Question: Can we see the original dot product in RBF kernel?

Assuming \mathbf{x} is a one-dimensional vector, we can rewrite the RBF kernel as:

$$K(x, z) = \exp(-\gamma \|x - z\|^2) = \exp(-\gamma (x - z)^2)$$

Expanding the squared term:

$$(x - z)^2 = x^2 - 2xz + z^2$$

Does RBF Involve Dot Product in Lower-Dimensional Space?

Question: Can we see the original dot product in RBF kernel?

Assuming \mathbf{x} is a one-dimensional vector, we can rewrite the RBF kernel as:

$$K(x, z) = \exp(-\gamma \|x - z\|^2) = \exp(-\gamma(x - z)^2)$$

Expanding the squared term:

$$(x - z)^2 = x^2 - 2xz + z^2$$

Substituting back into the RBF kernel:

$$\begin{aligned} K(x, z) &= \exp(-\gamma(x^2 - 2xz + z^2)) \\ &= \exp(-\gamma x^2) \cdot \exp(2\gamma xz) \cdot \exp(-\gamma z^2) \end{aligned}$$

Key insight: The middle term $\exp(2\gamma xz)$ contains the dot product xz from the original space!

SVM: Parametric vs Non-Parametric

Question: Is SVM parametric or non-parametric?

SVM: Parametric vs Non-Parametric

Question: Is SVM parametric or non-parametric?

Answer: It depends on the kernel!

- **Parametric:** Linear and polynomial kernels

SVM: Parametric vs Non-Parametric

Question: Is SVM parametric or non-parametric?

Answer: It depends on the kernel!

- **Parametric:** Linear and polynomial kernels
 - Fixed functional form

SVM: Parametric vs Non-Parametric

Question: Is SVM parametric or non-parametric?

Answer: It depends on the kernel!

- **Parametric:** Linear and polynomial kernels
 - Fixed functional form
 - Number of parameters independent of training data size

SVM: Parametric vs Non-Parametric

Question: Is SVM parametric or non-parametric?

Answer: It depends on the kernel!

- **Parametric:** Linear and polynomial kernels
 - Fixed functional form
 - Number of parameters independent of training data size
- **Non-parametric:** RBF kernel

SVM: Parametric vs Non-Parametric

Question: Is SVM parametric or non-parametric?

Answer: It depends on the kernel!

- **Parametric:** Linear and polynomial kernels
 - Fixed functional form
 - Number of parameters independent of training data size
- **Non-parametric:** RBF kernel
 - Model complexity grows with data

SVM: Parametric vs Non-Parametric

Question: Is SVM parametric or non-parametric?

Answer: It depends on the kernel!

- **Parametric:** Linear and polynomial kernels
 - Fixed functional form
 - Number of parameters independent of training data size
- **Non-parametric:** RBF kernel
 - Model complexity grows with data
 - Uses all support vectors for prediction

RBF is Non-Parametric

$$\begin{aligned}\hat{\mathbf{y}}(\mathbf{x}_{\text{test}}) &= \text{sign}(\mathbf{w} \cdot \mathbf{x}_{\text{test}} + b) \\ &= \text{sign}\left(\sum_{j=1}^{N_{\text{SV}}} \alpha_j y_j \mathbf{x}_j \cdot \mathbf{x}_{\text{test}} + b\right) \\ \hat{\mathbf{y}}(\mathbf{x}_{\text{test}}) &= \text{sign}\left(\sum_{j=1}^N \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_{\text{test}}) + b\right)\end{aligned}$$

$\alpha_j = 0$ where $j \neq \text{S.V.}$

Interpretation of RBF

- $\hat{\mathbf{y}}(\mathbf{x}) = \text{sign}(\sum \alpha_i y_i \exp(-\|\mathbf{x} - \mathbf{x}_i\|^2) + b)$

Interpretation of RBF

- $\hat{\mathbf{y}}(\mathbf{x}) = \text{sign}(\sum \alpha_i y_i \exp(-\|\mathbf{x} - \mathbf{x}_i\|^2) + b)$
- $-\|\mathbf{x} - \mathbf{x}_i\|^2$ corresponds to radial term

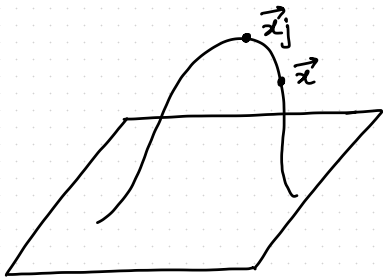
Interpretation of RBF

- $\hat{\mathbf{y}}(\mathbf{x}) = \text{sign}(\sum \alpha_i y_i \exp(-\|\mathbf{x} - \mathbf{x}_i\|^2) + b)$
- $-\|\mathbf{x} - \mathbf{x}_i\|^2$ corresponds to radial term
- $\sum \alpha_i y_i$ is the activation component

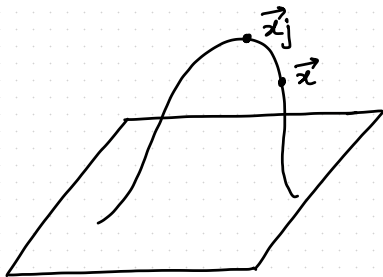
Interpretation of RBF

- $\hat{\mathbf{y}}(\mathbf{x}) = \text{sign}(\sum \alpha_i y_i \exp(-\|\mathbf{x} - \mathbf{x}_i\|^2) + b)$
- $-\|\mathbf{x} - \mathbf{x}_i\|^2$ corresponds to radial term
- $\sum \alpha_i y_i$ is the activation component
- $\exp(-\|\mathbf{x} - \mathbf{x}_i\|^2)$ is the basis component

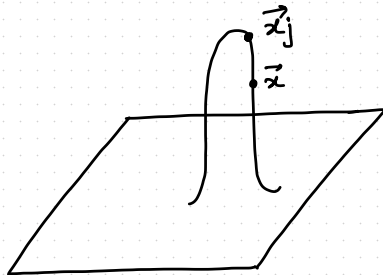
RBF INTERPRETATION



RBF INTERPRETATION



LOW γ
HIGH INFLUENCE OF \vec{x}_j



HIGH γ
LOW INFLUENCE OF \vec{x}_j

Summary

Key Takeaways

- **Goal:** SVM finds optimal separating hyperplane by maximizing margin

Key Takeaways

- **Goal:** SVM finds optimal separating hyperplane by maximizing margin
- **Math:** Dual formulation enables kernel trick

Key Takeaways

- **Goal:** SVM finds optimal separating hyperplane by maximizing margin
- **Math:** Dual formulation enables kernel trick
- **Power:** Kernels enable non-linear classification without explicit mapping

Key Takeaways

- **Goal:** SVM finds optimal separating hyperplane by maximizing margin
- **Math:** Dual formulation enables kernel trick
- **Power:** Kernels enable non-linear classification without explicit mapping
- **Popular kernels:** Linear, Polynomial, RBF (Gaussian)

Key Takeaways

- **Goal:** SVM finds optimal separating hyperplane by maximizing margin
- **Math:** Dual formulation enables kernel trick
- **Power:** Kernels enable non-linear classification without explicit mapping
- **Popular kernels:** Linear, Polynomial, RBF (Gaussian)
- **Remarkable:** RBF kernel \leftrightarrow infinite-dimensional space

Key Takeaways

- **Goal:** SVM finds optimal separating hyperplane by maximizing margin
- **Math:** Dual formulation enables kernel trick
- **Power:** Kernels enable non-linear classification without explicit mapping
- **Popular kernels:** Linear, Polynomial, RBF (Gaussian)
- **Remarkable:** RBF kernel \leftrightarrow infinite-dimensional space
- **Flexibility:** Parametric (linear/poly) or non-parametric (RBF)

Key Takeaways

- **Goal:** SVM finds optimal separating hyperplane by maximizing margin
- **Math:** Dual formulation enables kernel trick
- **Power:** Kernels enable non-linear classification without explicit mapping
- **Popular kernels:** Linear, Polynomial, RBF (Gaussian)
- **Remarkable:** RBF kernel \leftrightarrow infinite-dimensional space
- **Flexibility:** Parametric (linear/poly) or non-parametric (RBF)
- **Efficiency:** Only support vectors matter for prediction

Next Steps

- Soft-margin SVM for non-separable data

Next Steps

- Soft-margin SVM for non-separable data
- Hyperparameter tuning (C , γ)

Next Steps

- Soft-margin SVM for non-separable data
- Hyperparameter tuning (C , γ)
- Multi-class SVM extensions

Next Steps

- Soft-margin SVM for non-separable data
- Hyperparameter tuning (C , γ)
- Multi-class SVM extensions
- Computational considerations and optimization

Next Steps

- Soft-margin SVM for non-separable data
- Hyperparameter tuning (C , γ)
- Multi-class SVM extensions
- Computational considerations and optimization
- Comparison with other classifiers